App. No. 09/560,269
Amendment dated April 4, 2006
Reply to Office Action of January 4, 2006

Listing of claims:

1.   (Currently amended) A computer-implemented method for collecting information relating to execution of an application, the method being executed on a computer, the method comprising:

determining a set of probe locations in the application, wherein the set of probe locations comprises:

a probe location at a beginning of a calling function,

a probe location at an end of the calling function,

a probe location at a beginning of a first called function,

a probe location at an end of the first called function,

a probe location at a position in the calling function where the calling function calls the first called function,

a probe location at a position in the calling function where the first called function returns after execution,

a probe location at a beginning of a second called function when the first called function calls the second called function, and

when the first called function calls the second called function and when the second called function returns to the calling function, a probe location at a position in the calling function where the second called function returns after execution,

wherein a ~~probe location~~ pair of probe locations ~~that~~ produces redundant information;

eliminating one member of the probe location pair, ~~wherein the eliminated member comprises one of:~~

~~a call to function probe location in a calling function when the calling function calls to a called function, wherein the called function is within a current module of the calling function, and~~

~~a return from function probe location in the calling function when the calling function calls to the called function, wherein the called function is within the current module of the calling function;~~ and

App. No. 09/560,269
Amendment dated April 4, 2006
Reply to Office Action of January 4, 2006

inserting probes at the remaining probe locations in the application such that data collected relating to the execution of the application produces non-redundant information.

Claims 2-4   (Cancelled)

5.   (Currently amended) The method of claim 1 [[4]], wherein a ~~the~~ first probe is configured to collect an address of the a first called function ~~in which the identified first location is located~~, an address of the a second called function ~~in which the second location is located~~, a first stack pointer, and a first time indicator, and further wherein a ~~the~~ second probe is configured to collect the address of the second called function, a second stack pointer, and a second time indicator.

6.   (Cancelled)

7.   (Currently amended) The method of claim 1 ~~6~~, wherein a ~~the~~ first probe is configured to collect an address of the calling function, an address of the first called function, a first stack pointer, and a first time indicator, and further wherein a ~~the~~ second probe is configured to collect the address of the first called function, a second stack pointer, and a second time indicator.

8.   (Original)   The method of claim 1, further comprising:
        identifying a block of code to which execution of the application is directed upon occurrence of an error; and
        inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

9.   (Original) The method of claim 8, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

10.   (Original) The method of claim 1, further comprising using the inserted probes to collect the information relating to the execution of the application.

11.   (Original) The method of claim 10, further comprising analyzing the collected information.

12.   (Currently amended) A computer-implemented method for collecting information relating to execution of an application, the method being executed on a computer, the method comprising:

~~determining entry and exit points of a called function that is called from a calling function in the application;~~

determining a set of probe locations in the application, wherein the set of probe locations comprises:

a probe location at a beginning of a calling function,

a probe location at an end of the calling function,

a probe location at a beginning of a first called function,

a probe location at an end of the first called function,

a probe location at a position in the calling function where the calling function calls the first called function,

a probe location at a position in the calling function where the first called function returns after execution,

a probe location at a beginning of a second called function when the first called function calls the second called function, and

when the first called function calls the second called function and when the second called function returns to the calling function, a probe location at a position in the calling function where the second called function returns after execution,

wherein a pair of probe locations produces ~~location pairs that produce~~ redundant information;

eliminating one member of the ~~each~~ probe location pair~~, wherein the eliminated member comprises one of:~~

~~a call to function probe location in the calling function when the calling function calls to the called function, wherein the called function is within a current module of the calling function; and~~

~~a return from function probe location in the called function when the calling function calls to the called function, wherein the called function is within the current module of the calling function;~~

inserting probes at the remaining probe locations in the application;

collecting non-redundant information relating to the execution of the application using the inserted probes; and

analyzing the collected information.

13.   (Currently amended) The method of claim 12, ~~further comprising:~~

~~identifying a first location within the application at which a function call directs execution of the application to a second location outside of the current module; and~~

~~inserting a first probe before the identified first location and a second probe after the identified first location.~~

wherein a ~~the~~ first probe is configured to collect an address of ~~the~~ a first called function ~~in which the identified first location is located~~, an address of ~~the~~ a second called function ~~in which the second location is located~~, a first stack pointer, and a first time indicator, and further wherein a ~~the~~ second probe is configured to collect the address of the second called function, a second stack pointer, and a second time indicator.

14.   (Currently amended) The method of claim 12, ~~further comprising:~~

~~identifying a first location within a calling function at which execution of the application is directed to a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and~~

~~inserting a first probe before the first location and a second probe after the second location,~~

wherein a ~~the~~ first probe is configured to collect an address of the calling function, an address of the first called function, a first stack pointer, and a first time indicator,

Page 5 of 18

and <u>further wherein a</u> the second probe is configured to collect the address of the <u>first</u> called function, a second stack pointer, and a second time indicator.

15.   (Original) The method of claim 12, further comprising:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

16.   (Currently amended) A tangible computer-readable medium having an application including computer-executable instructions, the computer-executable instructions comprising:

determining a set of probe locations in the application, wherein the set of probe locations <u>comprises:</u>

> <u>a probe location at a beginning of a calling function,</u>
>
> <u>a probe location at an end of the calling function,</u>
>
> <u>a probe location at a beginning of a first called function,</u>
>
> <u>a probe location at an end of the first called function,</u>
>
> <u>a probe location at a position in the calling function where the calling</u>

<u>function calls the first called function,</u>

> <u>a probe location at a position in the calling function where the first called</u>

<u>function returns after execution,</u>

> <u>a probe location at a beginning of a second called function when the first</u>

<u>called function calls the second called function, and</u>

> <u>when the first called function calls the second called function and when</u>

<u>the second called function returns to the calling function, a probe location at a position in</u>

<u>the calling function where the second called function returns after execution,</u>

wherein ~~includes~~ a ~~probe-location~~ pair <u>of probe locations</u> ~~that~~ produces redundant information;

eliminating one member of the probe location pair, ~~wherein the eliminated member comprises one of:~~

~~a call-to-function-probe location in a calling function when the calling function calls to a called function, wherein the called function is within a current module of the calling function, and~~

~~a return-from-function probe-location in the calling function when the calling function calls to the called function, wherein the called function is within the current module of the calling function;~~ and

inserting probes at the remaining probe locations in the application such that data collected relating to the execution of the application produces non-redundant information.

Claims 17-19 (Cancelled)

20.   (Currently amended) The computer-readable medium of claim <u>16</u> ~~19~~, wherein <u>a</u> ~~the~~ first probe is configured to collect an address of <u>the</u> ~~a~~ first <u>called</u> function ~~in which the identified first location is located~~, an address of <u>the</u> ~~a~~ second <u>called</u> function ~~in which the second location is located~~, a first stack pointer, and a first time indicator, and <u>further wherein a</u> ~~the~~ second probe is configured to collect the address of the second <u>called</u> function, a second stack pointer, and a second time indicator.

21.   (Cancelled)

22.   (Currently amended) The computer-readable medium of claim <u>16</u> ~~21~~, wherein <u>a</u> ~~the~~ first probe is configured to collect an address of the calling function, an address of the <u>first</u> called function, a first stack pointer, and a first time indicator, and <u>further wherein a</u> ~~the~~ second probe is configured to collect the address of the <u>first</u> called function, a second stack pointer, and a second time indicator.

23.   (Original) The computer-readable medium of claim 16, having further computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

24.   (Original) The computer-readable medium of claim 23, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

25.   (Original) The computer-readable medium of claim 16, having further computer-executable instructions for using the inserted probes to collect the information relating to the execution of the application.

26.   (Original) The computer-readable medium of claim 25, having further computer-executable instructions for analyzing the collected information.

27.   (Currently amended) A tangible computer-readable medium having an application including computer-executable instructions, the computer-executable instructions comprising:

~~determining entry and exit points of a called function that is called from a calling function in the application;~~

determining a set of probe locations in the application, wherein the set of probe locations comprises:

a probe location at a beginning of a calling function,

a probe location at an end of the calling function,

a probe location at a beginning of a first called function,

a probe location at an end of the first called function,

Page 8 of 18

a probe location at a position in the calling function where the calling function calls the first called function,

a probe location at a position in the calling function where the first called function returns after execution.

a probe location at a beginning of a second called function when the first called function calls the second called function, and

when the first called function calls the second called function and when the second called function returns to the calling function, a probe location at a position in the calling function where the second called function returns after execution,

wherein a pair of probe locations ~~location pair that~~ produces redundant information;

eliminating one member of the probe location pair~~, wherein the eliminated member comprises one of:~~

~~a call to function probe location in a calling function when the calling function calls to a called function, wherein the called function is within a current module of the calling function, and~~

~~a return from function probe location in the calling function when the calling function calls to the called function, wherein the called function is within the current module of the calling function;~~ and

inserting probes at the remaining probe locations in the application;

collecting non-redundant information relating to the execution of the application using the inserted probes; and

analyzing the collected information.

28.   (Currently amended) The computer-readable medium of claim 27, ~~having further computer-executable instructions for:~~

~~identifying a first location within the application at which a function call directs execution of the application to a second location outside of the current module; and~~

~~inserting a first probe before the identified first location and a second probe after the identified first location.~~

wherein a the first probe is configured to collect an address of the a first called function in which the identified first location is located, an address of the a second called function in which the second location is located, a first stack pointer, and a first time indicator, and further wherein a the second probe is configured to collect the address of the second called function, a second stack pointer, and a second time indicator.

29.   (Currently amended) The computer-readable medium of claim 27, having further computer-executable instructions for:

identifying a first location within the calling function at which execution of the application is directed to the called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location,

wherein a the first probe is configured to collect an address of the calling function, an address of the first called function, a first stack pointer, and a first time indicator, and further wherein a the second probe is configured to collect the address of the first called function, a second stack pointer, and a second time indicator.

30.   (Original) The computer-readable medium of claim 27, having further computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

31.   (Currently amended) A computer system comprising a processor that is arranged to execute computer-executable instructions, the computer-executable instructions comprising:

determining a set of probe locations in the application, wherein the set of probe locations comprises:

a probe location at a beginning of a calling function,

a probe location at an end of the calling function,

a probe location at a beginning of a first called function,

a probe location at an end of the first called function,

a probe location at a position in the calling function where the calling function calls the first called function,

a probe location at a position in the calling function where the first called function returns after execution,

a probe location at a beginning of a second called function when the first called function calls the second called function, and

when the first called function calls the second called function and when the second called function returns to the calling function, a probe location at a position in the calling function where the second called function returns after execution,

wherein ~~includes~~ a ~~probe location~~ pair of probe locations ~~that~~ produces redundant information;

eliminating one member of the probe location pair, ~~wherein the eliminated member comprises one of:~~

~~a call-to-function probe location in a calling function when the calling function calls to a called function, wherein the called function is within a current module of the calling function, and~~

~~a return-from-function probe location in the calling function when the calling function calls to the called function;~~ and

inserting probes at the remaining probe locations in the application such that data collected relating to the execution of the application produces non-redundant information.

Claims 32-34 (Cancelled)

35.   (Currently amended) The computer system of claim 31 34, wherein a the first probe is configured to collect an address of the a first called function in which the identified first location is located, an address of the a second called function in which the second location is located, a first stack pointer, and a first time indicator, and further wherein a the second probe is configured to collect the address of the second called function, a second stack pointer, and a second time indicator.

36.   (Cancelled)

37.   (Currently amended) The computer system of claim 31 36, wherein a the first probe is configured to collect an address of the calling function, an address of the first called function, a first stack pointer, and a first time indicator, and further wherein a the second probe is configured to collect the address of the first called function, a second stack pointer, and a second time indicator.

38.   (Previously presented) The computer system of claim 31, further configured to execute computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

39.   (Previously presented) The computer system of claim 38, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

40.   (Previously presented) The computer of claim 31, further configured to execute computer-executable instructions for using the inserted probes to collect the information relating to the execution of the application.

41.   (Previously presented) The computer system of claim 40, further configured to execute computer-executable instructions for analyzing the collected information.

42.   (Currently amended) A computer system comprising a processor that is arranged to execute computer-executable instructions, the computer-executable instructions comprising
~~determining entry and exit points of a called function that is called from a calling function in;~~
    determining a set of probe locations in the application, wherein the set of probe locations comprises:
        a probe location at a beginning of a calling function,
        a probe location at an end of the calling function,
        a probe location at a beginning of a first called function,
        a probe location at an end of the first called function,
        a probe location at a position in the calling function where the calling function calls the first called function,
        a probe location at a position in the calling function where the first called function returns after execution,
        a probe location at a beginning of a second called function when the first called function calls the second called function, and
        when the first called function calls the second called function and when the second called function returns to the calling function, a probe location at a position in the calling function where the second called function returns after execution,
    wherein a pair of probe locations ~~location pair that~~ produces redundant information;
    eliminating one member of the probe location pair~~, wherein the eliminated member comprises one of:~~
        ~~a call to function probe location in the calling function when the calling function calls to the called function, wherein the called function is within a current module of the calling function, and~~

~~a return from function probe location in the calling function when the calling function calls to the called function, wherein the called function is within the current module of the calling function;~~

inserting probes at the remaining probe locations in the application;

collecting non-redundant information relating to the execution of the application using the inserted probes; and

analyzing the collected information.

43.   (Currently amended) The computer system of claim 42, ~~further configured to execute computer-executable instructions for:~~

~~identifying a first location within the application at which a function-call directs execution of the application to a second location outside of the current-module; and~~

~~inserting a first probe before the identified first location and a second probe after the identified first location,~~

wherein a ~~the~~ first probe is configured to collect an address of <u>the</u> ~~a~~ first <u>called</u> function ~~in which the identified first location is located~~, an address of <u>the</u> ~~a~~ second <u>called</u> function ~~in which the second location is located~~, a first stack pointer, and a first time indicator, and <u>further wherein a</u> ~~the~~ second probe is configured to collect the address of the second <u>called</u> function, a second stack pointer, and a second time indicator.

44.   (Currently amended) The computer system of claim 42, ~~further configured to execute computer-executable instructions for:~~

~~identifying a first location within the calling function at which execution of the application is directed to the called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and~~

~~inserting a first probe before the first location and a second probe after the second location,~~

wherein a ~~the~~ first probe is configured to collect an address of the calling function, an address of the <u>first</u> called function, a first stack pointer, and a first time indicator,

App. No. 09/560,269
Amendment dated April 4, 2006
Reply to Office Action of January 4, 2006

and further wherein a the second probe is configured to collect the address of the first called function, a second stack pointer, and a second time indicator.

45.   (Previously presented) The computer system of claim 42, further configured to execute computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

Claims 46-51 (Cancelled)